METHOD AND SYSTEM FOR NOISE REDUCTION IN AN IMAGE

Steve Zhihua Zeng

BACKGROUND

[0001] Video images, and especially analog video signals can be corrupted by varied types of temporal and spatial noise during acquisition, transmission, and recording of the image. Typical types of noise include thermal noise, single frequency modulation distortion noise, and impulse noise. Noise reduction techniques that apply linear or non-linear filters on video signals can reduce the amount of noise. One such technique is to apply a low-pass filter on the video signal. However, simple low-pass filtering tends to produce over-smoothed video that appears blurry. Other filters such as Wiener filters, Kalman filters are better at removing one or more of spatial noise or temporal noise but can be expensive in terms of implementation and device costs.

[0002] Therefore, a method of noise reduction that overcomes these problems would be useful.

Field of the Disclosure

[0003] The present disclosure relates to data processing, and more specifically to image and video processing.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] The present disclosure may be better understood, and its features and advantages made apparent to those skilled in the art by referencing the accompanying drawings.

[0005] FIG. 1 represents, in flow diagram form, a method in accordance with the present disclosure;

[0006] FIG. 2 represents, in block diagram form, a system in accordance with a specific embodiment of the present disclosure.

[0007] FIG. 3 represents Boolean edge data in accordance with a specific embodiment of the present disclosure;

[0008] FIG. 4 represents weighted edge data in accordance with a specific embodiment of the present disclosure;

[0009] FIG. 5 represents, in flow diagram form, a method of blending accordance with a specific embodiment of the present disclosure;

[0010] FIGs. 6 and 7 represent, in flow diagram form, methods in accordance with specific embodiments of the present disclosure;

[0011] FIG. 8 represents, in block diagram form, a system in accordance with the present disclosure; and

[0012] FIG. 9 represents an intermediate data in accordance with a specific embodiment of the present disclosure.

[0013] The use of the same reference symbols in different drawings indicates similar or identical items.

DESCRIPTION OF THE PREFERRED EMBODIMENT(S)

[0014] In a specific embodiment of the present disclosure, a source image is smoothed to create a smoothed image, and an edge detector is used to create an edge layer. A blending controller is used to control a blending between the source image and the smoothed image. The blended destination image maintains detail while eliminating unwanted noise. Specific implementations of the present disclosure are better understood with reference to FIGS. 1-8.

[0015] FIG. 1 illustrates in flow diagram form a method in accordance with the present disclosure. The method of FIG. 1 is discussed with reference to the system of FIG. 2,

which illustrates in block diagram form data flow through a system 100 in accordance with the present disclosure.

[0016] System 100 comprises noise filter 140, edge detector 150 and blending controller 160. In addition, system 100 includes memory (not specifically shown) for storing image information including source image 112, smoothed layer 120, edge layer 130, and destination layer 135. The layers are accessible by the noise filter 140, edge detector 150 and blending controller 160.

[0017] A first image layer is received at step 11. Referring to FIG. 2, source layer 117 is an example of such a first image layer. The source layer 117 is one of three component layers, along with layers 115 and 116, which make up the source image 112. Examples of types of component layers 115-117 include RGB component layers, YUV component layers, and component layers of any other color spaces. The system 100 receives source image 112 by receipt of a video stream or by accessing a memory location.

[0018] The source providing the source image can be a head end device, a TV receiver, a Video Cassette Recorder, a DVD (Digital Versatile Disk) player, or other video sources.

[0019] Upon receipt of the source image 112, each image layer 115-117 is processed independently. For purposes of discussion, data flow with respect to FIGs. 1 and 2 will be discussed with respect to one of the source layers, source layer 117. In addition, it will be appreciated that the layer information may be stored and processed as frames or partial frames, such as line buffers depending upon a systems specific implementation. For ease of discussion, it will be assumed the image information is stored as entire frames.

[0020] At step 12, a first edge layer is determined based on the first image layer. With reference to FIG. 2, source layer 117 is processed by the edge detector 150 to determine the edge layer 130.

[0021] The edge layer 130 comprises a plurality of memory locations corresponding to the plurality of pixels of an image. The edge layer 130 contains a pixel edge indicator for each pixel that indicates whether a pixel is associated with an edge of an image. In one

embodiment, a pixel edge indicator is a Boolean representation indicating the presence or absence of an edge. For example, a positive Boolean pixel edge indicator would indicate that a specific pixel is part of an image edge.

[0022] The edge detector 150 can detect edges by determining a gradient for each pixel location. For example, the horizontal and vertical gradient of a pixel can be calculated using the equations

[0023] $\text{Grad\_x} = p(i+1, j) - p(i-1, j)$; and

[0024] $\text{Grad\_y} = p(i, j+1) - p(i, j-1)$.

[0025] A rectangular-to-polar conversion, such as $M(i, j) = \text{SQRT}(\text{Grad\_x}^2 + \text{grad\_y}^2)$, can then be performed to get a magnitude of the edge. Alternatively, the square root operation can be removed and the magnitude compared to a predefined edge-level that controls what is considered an edge, and ultimately controls how many details will be preserved in a final destination layer. For example, if the magnitude is larger than the predefined value of the edge-level, then the corresponding pixel is said to be an edge pixel. In other embodiments, the edge detector 150 can determine the presence of an edge solely based on a pixel's horizontal or vertical edge component.

[0026] FIG. 3 represents a portion 212 of a Boolean edge layer that contains Boolean pixel edge indicators. The 10 by 10 matrix of FIG. 3 represents the top left-most pixels of an image layer. For purposes of illustration, a "1" value indicates that the pixel at that location has been identified as an edge pixel, while a "0" value indicates that the pixel at that location has been identified as not being an edge pixel. In an alternate embodiment, the edge layer 130 can contain weighted edge indicators, as illustrated in FIG. 4.

[0027] FIG. 4 illustrates a portion of a weighted edge layer 214 that contains weighted edge indicators. The specific implementation of FIG. 4 assigns a weighted value to a pixel by determining a number of Boolean pixel edge indicators within +/- 1 pixels of the pixel and storing this value in the lower four-bits of a data byte, and storing in the upper four-bits of the data byte the number of Boolean pixel edge indicators that are at +/- two

- 4 -

pixels from the pixel. The number of Boolean pixel edge indicators in layer 212 that are at +/- two pixels from the pixel can be calculated by determining the number Boolean pixel edge indicators that are within +/- two pixels from the pixel and subtracting the number of Boolean pixels edge indicators that are +/- 1 pixel from the pixel.

[0028] For example, the number of Boolean edge pixels in layer 212 within +/- pixels of pixel P94,4) is indicated by the value 5 in layer 114. This value is stored in the lower 4-bits of the pixel P(4,4) in the weighted edge layer 214. The number of edge pixels within the box 203 of Boolean edge layer 212 is 12. The number of these pixels that are two pixels away from the pixel location P(4,4) is determined by subtracting the number of edge pixels within the box 201 from the edge pixels within the box 212, which results in a value of 7. This value is stored in the upper 4-bits of pixel P(4,4) in layer 211. Therefore the weighted pixel value of pixel P(4,4) is "75". It will be appreciated that many other schemes for determining and/or storing weighted edge values are possible.

[0029] Returning to FIG. 1, at step 13 the first image layer is blended with a first other layer based upon the first edge layer. In one embodiment, the first other layer is the source layer 117.

[0030] The blending controller 160 of FIG. 2 is used to implement the blending, and uses information of edge layer 130 to blend the source layer 117 with the smoothed layer 120 to preserve edges and fine structures in the source image layer 117. When the edge layer contains only Boolean edge information one of two blending ratios can be implemented by the blending controller 160 at each pixel location. However, typically, the ability to blend a pixel based on one of only two blending ratios will not provide enough blending options to provide a destination image with an enhanced image.

[0031] To provide additional levels of blending, a weighted edge layer, such as is illustrated in FIG. 4 can be used by the blending controller 160 to select one of more than two blending ratios with respect to the blending of a specific pixel. FIG. 5 discloses a specific blending method for use with weighted edge values.

[0032] At step 302, a determination is made whether the pixel edge value labeled WEIGHT1 is greater than a variable T1. WEIGHT1 represent the number of edge pixels within +/- 1 pixel of the pixel being scaled. With respect to FIG. 4, this would be the value stored at the lower 4 bits of a pixel location. If WEIGHT1 is greater than threshold T1, flow proceeds to step 322, where the source image pixel is copied directly to the destination layer, such as destination layer 135. Otherwise, flow proceeds to step 303.

[0033] At step 303, a determination is made whether the pixel edge value WEIGHT1 is greater than T2. If so, flow proceeds to step 323, where the source image pixel is blended with the smoothed image at a ratio of 3:1. Otherwise, flow proceeds to step 304.

[0034] At step 304, a determination is made whether the pixel edge value WEIGHT1 is greater than T3. If so, flow proceeds to step 3:1, where the source image pixel is blended with the smoothed image at a ratio of 1:1. Otherwise, flow proceeds to step 305.

[0035] At step 305, a determination is made whether the pixel edge value labeled WEIGHT2 is greater than a variable T4. WEIGHT2 represent the number of edge pixels at +/- 2 pixels of the pixel being scaled. With respect to FIG. 4, this would be the value stored at the upper 4 bits of a pixel location. If WEIGHT2 is greater than the value of X4 flow proceeds to step 325, where the source image pixel is blended with the smoothed image at a ratio of 1:3. Otherwise, flow proceeds to step 326.

[0036] At step 326, the destination pixel is set equal to the smoothed pixel.

[0037] At step 306, a determination is made whether there is another pixel. If so, flow proceeds to step 302 to process the next pixel, otherwise the flow ends.

[0038] The variables T1, T2, T3, and T4 are predetermined, and as such can be preset or user defined variables. In another embodiment, the variables T1 through T4 can be statistically determined based upon the source image. In a specific embodiment, the variables T1 to T4 are set to 7, 3, 1, and 3 respectively.

[0039] FIG. 6 illustrates another method in accordance with an embodiment of the present disclosure.

- 6 -

[0040] At step 21, a first image layer of an image is received. In a manner similar to step 11 of FIG. 1.

[0041] At step 22, the first other layer is determined. Typically, the first other layer is determined by noise filter 140, which filters the source layer to provide a smoothed image. Noise filter 140 can be any type of noise filter, but will typically be either a low-pass filter or median filter depending upon the cost-performance ratio considerations of system 100. In one embodiment, a low-pass filter consisting of a five-tap horizontal filter and a five-tap vertical filter is used. Different coefficients can be used depending upon a desired noise level. In one embodiment, three noise levels implemented by the noise filters have cut-off frequencies of 0.7fs, 0.5fs, and 0.3fs. An intermediate smoothing layer can be formed by applying the low-pass filter on the horizontal direction and storing the results in memory, with a final smoothed layer including filtering in the vertical direction being formed prior to blending. A 2-dimensional median filter can be used supporting three sizes: 1x1, 3x3, and 5x5.

[0042] At step 23, a first edge layer is determined in a manner similar to that discussed with respect to step 12 of FIG. 1.

[0043] At step 24, the first image and the first other layer are blended in a manner similar to that discussed with respect to step 13 of FIG. 1.

[0044] FIG. 7 illustrates another method in accordance with an embodiment of the present disclosure. Steps 41, 42, and 43 are similar in function to steps 21, 22, and 23 previously discussed.

[0045] Step 44 is similar to step 41 but receives a second source image layer instead of the first source image layer.

[0046] Step 45 is similar to step 42, but a second edge layer based on the second source image layer is determined.

[0047] Step 46 is similar to step 43, but the second source image layer is blended with the second other layer instead of the first source image layer being blended with the first layer. The result of step 46 is a second blended video layer.

[0048] At step 47, a composite image combining the first and second blended video layers is provided. It will be appreciated that typically, additional steps, analogous to steps 41-43 will be performed to generate a third blending layer from which a composite image is formed.

[0049] Figure 8 illustrates, in block diagram form, a data processing system that may represent a general purpose processing system, such as a personal computer or a personal digital assistant, or an application specific system such as a media server, internet appliance, home networking hubs, and the like. The system 500 is illustrated to include a central processing unit 510, which may be a conventional or proprietary data processor, memory including random access memory 512, read only memory 514, input output adapter 522, a user interface adapter 520, a communications interface adapter 524, and a multimedia controller 526.

[0050] The input output (I/O) adapter 526 can be further connected to various peripherals such as disk drives 547, printer 545, removable storage devices 546, as well as other standard and proprietary I/O devices.

[0051] The user interface adapter 520 can be considered to be a specialized I/O adapter. The adapter 520 is illustrated to be connected to a mouse 540, and a keyboard 541. In addition, the user interface adapter 520 may be connected to other devices capable of providing various types of user control, such as touch screen devices.

[0052] The communications interface adapter 524 is connected to a bridge 550 such as is associated with a local or a wide area network, and a modem 551. By connecting the system bus 502 to various communication devices, external access to information can be obtained.

[0053] The multimedia controller 526 will generally include a video graphics controller capable of generating smoothed images in the manner discussed herein that can be displayed, saved, or transmitted. In a specific embodiment illustrated the multimedia controller 526 can include a system of FIG. 2, which can be implemented in hardware or software. Software implementations can be stored in any on of various memory locations, including RAM 512 and ROM 514, in addition software implementation software can be stored in the multimedia controller 526. When implemented in software, the system of FIG. 2 may be a data processor within the controller 526 for executing instruction, or it maybe a shared processor, such as CPU 510.

[0054] The preceding detailed description of the figures, reference has been made to the accompanying drawings which form a part thereof, and to which show by way of illustration specific embodiments in which the invention may be practiced. It will be appreciated that many other varied embodiments that incorporate the teachings herein may be easily constructed by those skilled in the art. For example, intermediate edge layers can be used to derive the final edge layer used by the blending controller 160. Once such intermediate layer would contain pixel information that indicates horizontally adjacent pixel information. For example, FIG. 9 illustrates an intermediate table, where each pixel stores the number of edge pixels within +/- 1 horizontal pixel in the lower four-bits of the byte, and the number of edge pixels at +/- 2 horizontal pixels in the upper four-bits of the byte. Such an intermediate layer allows for efficient calculation of the final edge layer. For example, the number of edge pixels within +/- pixels for a pixel $P(x, y)$ can be determined by adding the lower four bits of pixels $P(x, y-1)$, $P(x,y)$, and $P(x, y+1)$. In a similar manner, the number of edge pixels at +/- 2 pixels for a pixel $P(x, y)$ is determined by adding the upper four bits of $P(x, y-2)$, $P(x, y-1)$, $P(x,y)$, $P(x, y+1)$, and $P(x, y+2)$ to the lower four bits of $P(x, y-2)$ and $P(x, y+2)$. Utilizing an intermediate layer in this fashion reduces the computations needed to calculate the weighted edge values by reusing the horizontal edge data. Accordingly, the present disclosure is not intended to be limited to the specific form set forth herein, but on the contrary, it is intended to cover such alternatives, modifications, and equivalents, as can be reasonably included within the spirit and scope of the invention. The preceding detailed description

- 9 -

is, therefore, not to be taken in a limiting sense, and the scope of the present disclosure is defined only by the appended claims.